LINUX LAB – PRACTICE SESSION

Name : Reshma.P
Reg no: 17MIS1009

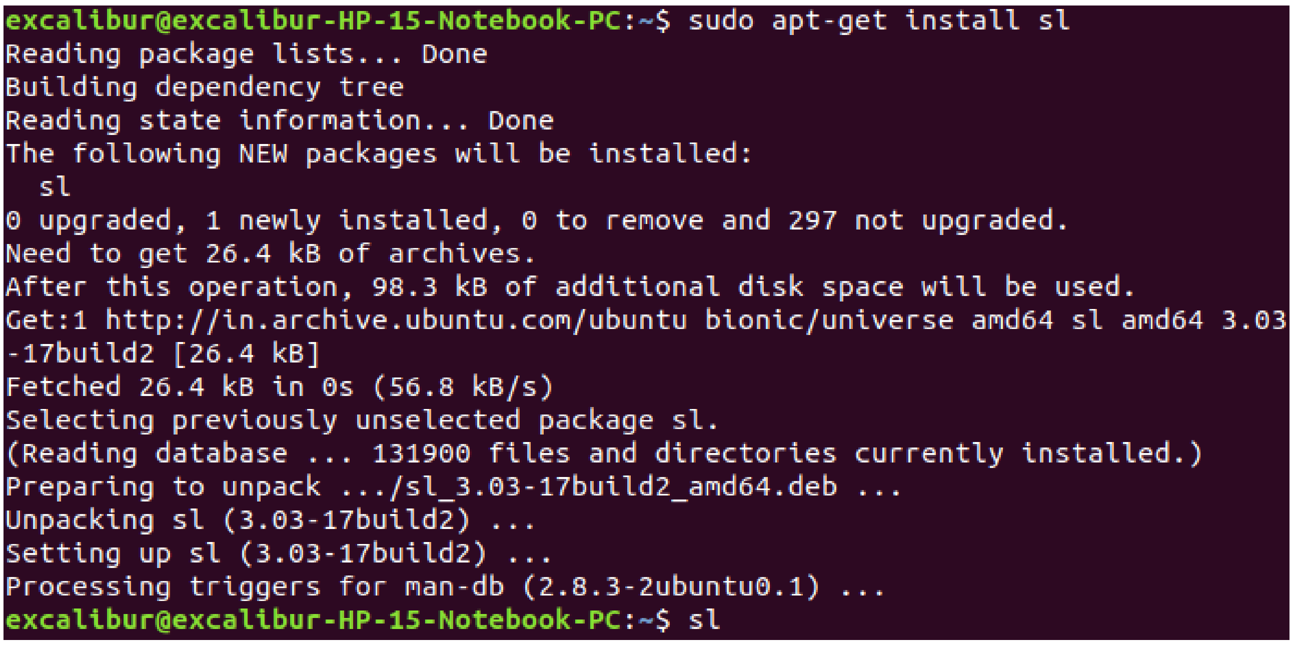
1) script 1:
File name : sl.sh

code:

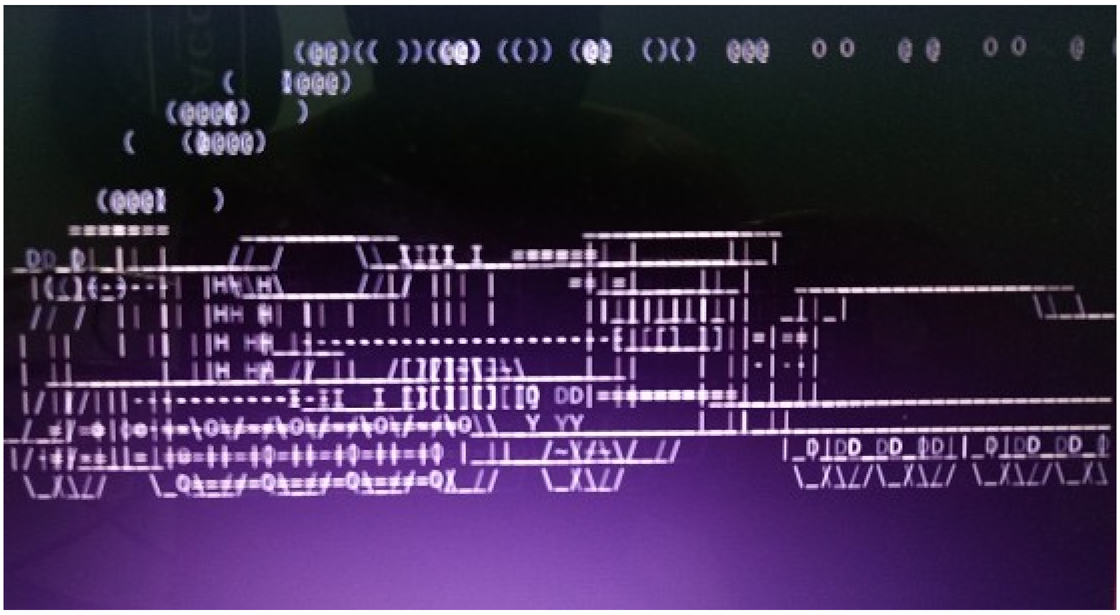
```
 sudo apt-get install sl
sl
```

EXPLANATION:
sl is just a fun command when you install the package and execute sl in terminal a train like structure will move in your terminal.

OUTPUT:



```
excalibur@excalibur-HP-15-Notebook-PC:~$ sudo apt-get install sl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sl
0 upgraded, 1 newly installed, 0 to remove and 297 not upgraded.
Need to get 26.4 kB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 sl amd64 3.03
-17build2 [26.4 kB]
Fetched 26.4 kB in 0s (56.8 kB/s)
Selecting previously unselected package sl.
(Reading database ... 131900 files and directories currently installed.)
Preparing to unpack .../sl_3.03-17build2_amd64.deb ...
Unpacking sl (3.03-17build2) ...
Setting up sl (3.03-17build2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
excalibur@excalibur-HP-15-Notebook-PC:~$ sl
```



---

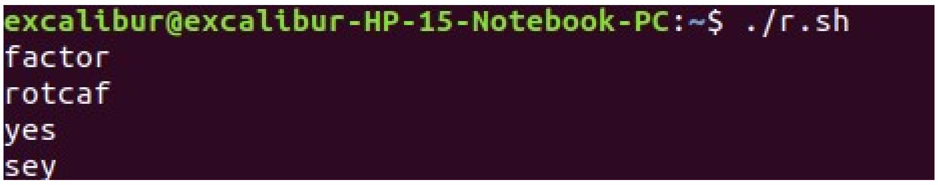
2) Script 2

File name : r.sh

CODE:
rev

EXPLANATION:
the command “rev” is used to reverse the lines. Once it is executed it will prompt for you to enter any word and when you press enter the word will be reversed and displayed.

output:



```
excalibur@excalibur-HP-15-Notebook-PC:~$ ./r.sh
factor
rotcaf
yes
sey
```

---

3) additional question: C program to implement simple reader writer with shared memory segment with semaphore

File name: op.c

code:

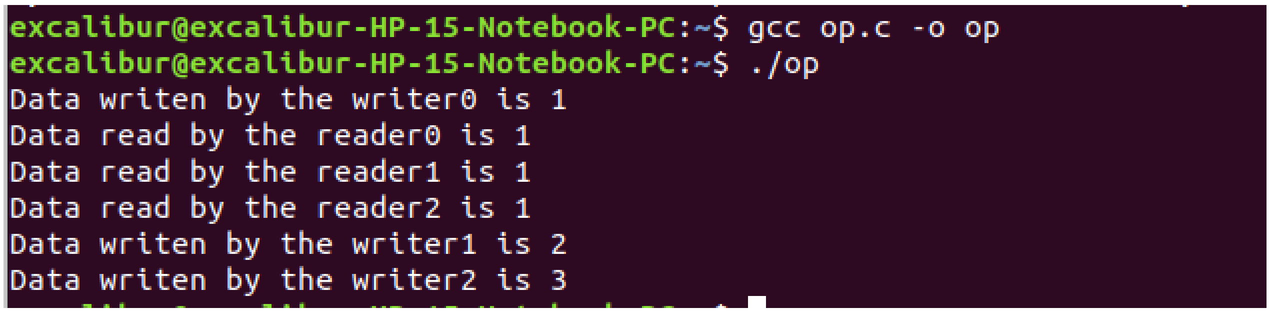
```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
```

```
sem_t mutex,writeblock;
int data = 0,rcount = 0;

void *reader(void *arg)
{
  int f;
  f = ((int)arg);
  sem_wait(&mutex);
  rcount = rcount + 1;
  if(rcount==1)
   sem_wait(&writeblock);
  sem_post(&mutex);
  printf("Data read by the reader%d is %d\n",f,data);
  sleep(1);
  sem_wait(&mutex);
  rcount = rcount - 1;
  if(rcount==0)
   sem_post(&writeblock);
  sem_post(&mutex);
}

void *writer(void *arg)
{
  int f;
  f = ((int) arg);
  sem_wait(&writeblock);
  data++;
  printf("Data writen by the writer%d is %d\n",f,data);
  sleep(1);
  sem_post(&writeblock);
}

main()
{
  int i,b;
  pthread_t rtid[5],wtid[5];
  sem_init(&mutex,0,1);
  sem_init(&writeblock,0,1);
  for(i=0;i<=2;i++)
  {
    pthread_create(&wtid[i],NULL,writer,(void *)i);
    pthread_create(&rtid[i],NULL,reader,(void *)i);
  }
  for(i=0;i<=2;i++)
  {
    pthread_join(wtid[i],NULL);
    pthread_join(rtid[i],NULL);
  }
}
```



```
excalibur@excalibur-HP-15-Notebook-PC:~$ gcc op.c -o op
excalibur@excalibur-HP-15-Notebook-PC:~$ ./op
Data writen by the writer0 is 1
Data read by the reader0 is 1
Data read by the reader1 is 1
Data read by the reader2 is 1
Data writen by the writer1 is 2
Data writen by the writer2 is 3
```

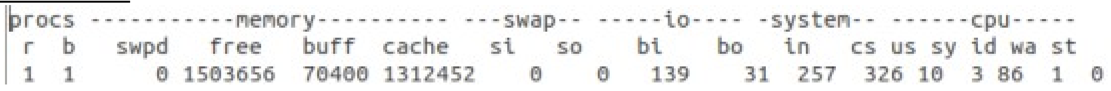
---

4) HOT question : Write a bash script to monitor health of the system.

File name : healthreport.sh:

```
vmstat 1200 > vmstat1.data
filename= "/home/excalibur/vmstat1.data"
tail -f $filename |
while read $line do
if [ (cat vmstat1.data | grep "swap")>0  ]
then
 echo "some rogue process has consumed massive amounts of memory"> swap.txt
fi
if [ (cat vmstat1.data | grep "r")>1  ]
then
 echo "some process are waiting to execute"> runqueue.txt
fi
if [ (cat vmstat1.data | grep "cpu")>1000  ]
then
 echo "cpu usage is more"> cpu.txt
fi
End
```

vmstat1.data



```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  1      0 1503656  70400 1312452    0    0   139    31  257  326 10  3 86  1  0
```

EXPLANATION:

the vmstat 1200 – monitors every 24 hours and puts the data into the vmstat1.data

grep “swap”- the swap should always be zero if its not then some process has consumed massive memory. That will be monitored in this line

grep “r”- the running queue is constantly above process 1 it indicates the system is slow and some process is waiting to be executed. That will be monitored here.

Grep “cpu”- it indicates the cpu usage of the system. If the cpu usage is more it will be monitored and will alert in this line.

GITHUB LINK:

https://github.com/deepthought101/Linux-lab/tree/master